

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A method for dynamically developing a user interface in an existing software application, comprising:

invoking a user interface developer component for creating the user interface during the execution of the software application from within the software application, the user interface developer component being implemented as part of the software application;

identifying one or more fields to include in the user interface;

associating a field type for each of the identified one or more fields;

associating the user interface with a function of the software application;

saving the identified one or more fields, associated field types, and the association of the user interface to the function of the software application in a user interface definition file; and

generating the user interface when the associated function is triggered based on the user interface definition file during the execution of the software application.

2. (Original) A method according to claim 1, further comprising:
providing one or more values for at least one of the identified one or more fields depending upon the associated field type; and
saving the one or more values in the user interface definition file.

3. (Original) A method according to claim 1, wherein the user interface definition file is saved as an XML file.

4. (Original) A method according to claim 1, wherein the generating includes parsing the user interface definition file to generate the user interface.

5. **(Original)** A method according to claim 4, wherein the generating further includes transforming the parsed user interface definition file into one or more objects.

6. **(Original)** A method according to claim 5, wherein the one or more objects are Java objects.

7. **(Original)** A method according to claim 5, wherein the generating further includes displaying the user interface based on the one or more objects.

8. **(Original)** A method according to claim 1, wherein the user interface developer component is implemented as a plug-in for the software application.

9. **(Currently Amended)** A software application operable on a computer system having a user interface developer component for dynamically developing a user interface for the software application, the software application configured to:

invoke the user interface developer component for creating the user interface during the execution of the software application from within the software application, the user interface developer component being implemented as part of the software application;

identify one or more fields to include in the user interface;

associate a field type for each of the identified one or more fields;

associate the user interface with a function of the software application;

save the identified one or more fields, associated field types, and the associated function in a user interface definition file; and

generate the user interface when the associated function is triggered based on the user interface definition file during the execution of the software application.

10. **(Original)** A software application according to claim 9, further configured to:
provide one or more values for at least one of the identified one or more fields depending upon the associated field type; and
save the one or more values in the user interface definition file.

11. **(Original)** A software application according to claim 9, wherein the user interface definition file is saved as an XML file.

12. **(Original)** A software application according to claim 9, further configured to parse the user interface definition file to generate the user interface.

13. **(Original)** A software application according to claim 12, further configured to transform the parsed user interface definition file into one or more objects.

14. **(Original)** A software application according to claim 13, wherein the one or more objects are Java objects.

15. **(Original)** A software application according to claim 13, further configured to display the user interface based on the one or more objects.

16. **(Original)** A software application according to claim 9, wherein the user interface developer component is implemented as a plug-in for the software application.

17. **(Currently Amended)** A computer system for dynamically developing a user interface for a software application, comprising:

a processor; and

a memory, coupled to the processor, comprising a plurality of instructions executed by the processor, the plurality of instructions configured to:

invoke a user interface developer component for creating the user interface during the execution of the software application from within the software application, the user interface developer component being implemented as part of the software application;

identify one or more fields to include in the user interface;

associate a field type for each of the identified one or more fields;

associate the user interface with a function of the software application;

save the identified one or more fields, associated field types, and the association of the user interface to the function of the software application in a user interface definition file; and

generate the user interface when the associated function is triggered based on the user interface definition file during the execution of the software application.

18. (Original) A computer system according to claim 17, the memory further comprising instructions configured to:

provide one or more values for at least one of the identified one or more fields depending upon the associated field type; and

save the one or more values in the user interface definition file.

19. (Original) A computer system according to claim 17, wherein the user interface definition file is saved as an XML file.

20. (Original) A computer system according to claim 17, the memory further comprising an instruction configured to parse the user interface definition file to generate the user interface.

21. (Original) A computer system according to claim 20, the memory further comprising an instruction configured to transform the parsed user interface definition file into one or more objects.

22. (Original) A computer system according to claim 21, wherein the one or more objects are Java objects.

23. (Original) A computer system according to claim 21, the memory further comprising an instruction configured to display the user interface based on the one or more objects.

24. (Original) A computer system according to claim 17, wherein the user interface developer component is implemented as a plug-in for the software application.

25. (Currently Amended) A computer readable medium on a computer system having a user interface developer component for dynamically developing a user interface in a software application, the computer readable medium configured to:

invoke the user interface developer component for creating the user interface during the execution of the software application from within the software application, the user interface developer component being implemented as part of the software application;

identify one or more fields to include in the user interface;

associate a field type for each of the identified one or more fields;

associate the user interface with a function of the software application;

save the identified one or more fields, associated field types, and the association of the user interface to the function of the software application in a user interface definition file; and

generate the user interface when the associated function is triggered based on the user interface definition file during the execution of the software application.

26. (Original) A computer readable medium according to claim 25, further configured to:

provide one or more values for at least one of the identified one or more fields depending upon the associated field type; and

save the one or more values in the user interface definition file.

27. (Original) A computer readable medium according to claim 25, wherein the user interface definition file is saved as an XML file.

28. (Original) A computer readable medium according to claim 25, further configured to parse the user interface definition file to generate the user interface.

29. (Original) A computer readable medium according to claim 28, further configured to transform the parsed user interface definition file into one or more objects.

30. (Original) A computer readable medium according to claim 29, wherein the one or more objects are Java objects.

31. **(Original)** A computer readable medium according to claim 29, further configured to display the user interface based on the one or more objects.

32. **(Original)** A computer readable medium according to claim 25, wherein the user interface developer component is implemented as a plug-in for the software application.

33. **(Currently Amended)** A system for dynamically developing a user interface in an existing software application, comprising:

means for invoking a user interface developer component for creating the user interface during the execution of the software application from within the software application, the user interface developer component being implemented as part of the software application;

means for identifying one or more fields to include in the user interface;

means for associating a field type for each of the identified one or more fields;

means for associating the user interface with a function of the software application;

means for saving the identified one or more fields, associated field types, and the association of the user interface to the function of the software application in a user interface definition file; and

means for generating the user interface when the associated function is triggered based on the user interface definition file during the execution of the software application.

34. **(Original)** A system according to claim 33, further comprising:

means for providing one or more values for at least one of the identified one or more fields depending upon the associated field type; and

means for saving the one or more values in the user interface definition file.

35. **(Original)** A system according to claim 33, wherein the user interface definition file is saved as an XML file.

36. **(Original)** A system according to claim 33, wherein the means for generating includes means for parsing the user interface definition file to generate the user interface.

37. **(Original)** A system according to claim 36, wherein the means for generating further includes means for transforming the parsed user interface definition file into one or more objects.

38. **(Original)** A system according to claim 37, wherein the one or more objects are Java objects.

39. **(Original)** A system according to claim 37, wherein the means for generating further includes means for displaying the user interface based on the one or more objects.

40. **(Original)** A system according to claim 33, wherein the user interface developer component is implemented as a plug-in for the software application.

41. **(Previously Presented)** A method according to claim 1, wherein the associated function is triggered in response to an input received by the software application.

42. **(Previously Presented)** A method according to claim 41, wherein the input is a selection from a drop down menu of the software application.

43. **(Previously Presented)** A method according to claim 1, wherein the step of associating the user interface with a function includes receiving a selection of the function from a list of functions of the software application.